Efficient Diversification of Web Search Results

G. Capannini, F. M. Nardini, R. Perego, and F. Silvestri
ISTI-CNR, Pisa, Italy
Web Search Results Diversification

• Query: “Vinci”, what is the user’s intent?
  • Information on Leonardo da Vinci?
  • Information on Vinci, the small village in Tuscany?
  • Information on Vinci, the company?
  • Others?
Web Search Results Diversification

- What is the user's intent?

**Vinci**

Leonardo da Vinci, the company?

Vinci is a French construction and engineering company, formerly known as Société Générale d'Entreprises. It employs over 165,000 people and is the largest construction company...

Vinci (construction) - Wikipedia, the free encyclopedia

History | Ownership | Financial data | Turnover analysis

Vinci is a French construction and electrical engineering company, formerly called Société Générale d'Entreprises. It employs over 165,000 people and is the largest construction company...

Vinci, manufacturer of high-quality baseball gloves, softball gloves and equipment since 1957, was founded on a promise to a father that, "One day your..."
Results Diversification as a Coverage Problem

• Hypothesis:
  • For each user’s query I can tell what is the set of all possible intents
  • For each document in the collection I can tell what are all the possible user’s intents it represents
  • each intent for each document is, possibly, weighted by a value representing how much that intent is represented by that document (e.g., 1/2 of document D is related to the intent of “digital photography techniques”)

• Goal:
  • Select the set of $k$ documents in the collection covering the maximum amount of intent weight. i.e., maximize the number of satisfied users.
State-of-the-Art Methods

• IASelect:

• xQuAD:
Diversify\((k)\)

DIVERSIFY\((k)\): Given query \(q\), a set of documents \(R_q\), a probability distribution of categories for the query \(P(c|q)\), the quality values of the documents \(V(d|q,c)\), \(\forall d \in D\) and an integer \(k\). Find a set of documents \(S \subseteq R_q\) with \(|S| = k\) that maximizes

\[
P(S|q) = \sum_{c} P(c|q) \left( 1 - \prod_{d \in S} (1 - V(d|q,c)) \right)
\]
Diversify\((k)\)

\textit{DIVERSEY}(k): Given query \(q\), a set of documents \(R_q\), a probability distribution of categories for the query \(P(c|q)\), the quality values of the documents \(V(d|q,c)\), \(\forall d \in D\) and an integer \(k\). Find a set of documents \(S \subseteq R_q\) with \(|S| = k\) that maximizes

\[
P(S|q) = \sum_c P(c|q) \left( 1 - \prod_{d \in S} (1 - V(d|q,c)) \right)
\]
Diversity(k): Given query q, a set of documents $R_q$, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q, c)$, $\forall d \in D$ and an integer $k$. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$P(S|q) = \sum_c P(c|q) \left( 1 - \prod_{d \in S} (1 - V(d|q, c)) \right)$$
Diversify($k$)

**DIVERSIFY($k$):** Given query $q$, a set of documents $R_q$, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in D$ and an integer $k$. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$P(S|q) = \sum_c P(c|q) \left( 1 - \prod_{d \in S} (1 - V(d|q,c)) \right)$$

- **Intents**
- **Weight**
- **Is the probability of being relative to intent c**
**Diversify**\((k)\)

**DIVERSIFY**\((k)\): Given query \(q\), a set of documents \(R_q\), a probability distribution of categories for the query \(P(c|q)\), the quality values of the documents \(V(d|q,c)\), \(\forall d \in D\) and an integer \(k\). Find a set of documents \(S \subseteq R_q\) with \(|S| = k\) that maximizes

\[
P(S|q) = \sum_c P(c|q) \left( 1 - \prod_{d \in S} (1 - V(d|q,c)) \right)
\]
Diversify($k$)

**DIVERSIFY**($k$): Given query $q$, a set of documents $R_q$, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in D$ and an integer $k$. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$P(S|q) = \sum_c P(c|q) \left( 1 - \prod_{d \in S} (1 - V(d|q,c)) \right)$$
Diversify($k$)

DIVERSIFY($k$): Given query $q$, a set of documents $R_q$, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in D$ and an integer $k$. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$P(S|q) = \sum_c P(c|q) \left( 1 - \prod_{d \in S} (1 - V(d|q,c)) \right)$$

Where:
- $P(c|q)$ is the probability of being relative to intent $c$.
- The weight is the probability of being relative to intent $c$.
- At least one doc is pertinent to $c$.
- No doc is pertinent to $c$.
- $d$ is not pertinent to $c$. 
- $d$ is not pertinent to $c$. 
- Pertinent to $c$. 
- Pertinent to $c$. 
- Pertinent to $c$. 
- Pertinent to $c$. 
- Pertinent to $c$. 

Known Results

• Diversify($k$) is NP-hard:
  • Reduction from max-weight coverage
• Diversify($k$)’s objective function is sub-modular:
  • Admits a $(1 - 1/e)$-approx. algorithm.
  • The algorithm works by inserting one result at a time, we insert the result with the max marginal utility.
• Quadratic complexity in the number of results to consider:
  • at each iteration scan the complete list of not-yet-inserted results.
Known Results

- Diversify\((k)\) is NP-hard.
- Reduction from max-weight coverage.
- Diversify\((k)\)’s objective function is sub-modular:
  - Admits a \((1-1/e)\)-approx. algorithm.
  - The algorithm works by inserting one result at a time, we insert the result with the max marginal utility.
  - Quadratic complexity in the number of results to consider:
    - at each iteration scan the complete list of not-yet-inserted results.

**Definition 1 (Submodularity).** Given a finite ground set \(N\), a set function \(f : 2^N \rightarrow \mathbb{R}\) is submodular if and only if for all sets \(S, T \subseteq N\) such that \(S \subseteq T\), and \(d \in N \setminus T\),
\[
f(S + d) - f(S) \geq f(T + d) - f(T).
\]
It looks reasonable, but...

- ... it may not diversify!
- The objective function is NOT about including as many categories as possible in the final results set.
- It is possible that even if there are less than $k$ categories, NOT all categories will be covered:
  - the formulation explicitly considers how well a document satisfies a given category.
- If a category $c$ is dominant and not well satisfied, more documents from $c$ will be added:
  - possible at the expense of not showing certain categories altogether.
xQuAD_Diversify\( (k) \):

Given a query \( q \), a set of ranked documents \( R_q \) retrieved for \( q \), a mixing parameter \( \lambda \in [0, 1] \), two probability distributions \( P(d|q) \) and \( P(d, \bar{S}|q) \) measuring, respectively, the likelihood of document \( d \) being observed given \( q \), and the likelihood of observing \( d \) but not the documents in the solution \( S \). Find a set of documents \( S \subseteq R_q \) with \( |S| = k \) that maximizes for each \( d \in S \)

\[
(1 - \lambda) \cdot P(d|q) + \lambda \cdot P(d, \bar{S}|q)
\]
xQuAD\_Diversify(k): Given a query $q$, a set of ranked documents $R_q$ retrieved for $q$, a mixing parameter $\lambda \in [0, 1]$, two probability distributions $P(d|q)$ and $P(d, \bar{S}|q)$ measuring, respectively, the likelihood of document $d$ being observed given $q$, and the likelihood of observing $d$ but not the documents in the solution $S$. Find a set of documents $S \subseteq R_q$ with $d \in S$

\[
P(d, \bar{S}|q) = \sum_{q' \in S_q} \left[ P(q'|q) P(d|q') \prod_{d_j \in S} 1 - P(d_j|q') \right]
\]

\[
(1 - \lambda) \cdot P(d|q) + \lambda \cdot P(d, \bar{S}|q)
\]
xQuAD_Diversify\((k)\): Given a query \(q\), a set of ranked documents \(R_q\) retrieved for \(q\), a mixing parameter \(\lambda \in [0, 1]\), two probability distributions \(P(d|q)\) and \(P(d, \bar{S}|q)\) measuring, respectively, the likelihood of document \(d\) being observed given \(q\), and the likelihood of observing \(d\) but not the documents in the relative \(S\). Find a set of documents \(S \subseteq R_q\) with

\[
P(d, \bar{S}|q) = \sum_{q' \in S_q} \left[ P(q'|q) P(d|q') \prod_{d_j \in S} 1 - P(d_j|q') \right]
\]

\[
(1 - \lambda) \cdot P(d|q) + \lambda \cdot P(d, \bar{S}|q)
\]

Same problem as before...
It may not diversify!
Our Proposal: MaxUtility
Our Proposal: MaxUtility
Our Proposal: MaxUtility
Our Proposal: MaxUtility
Our Proposal: MaxUtility

Leonardo da Vinci
Vinci
Vinci Town
Vinci Group

\( R_q \)

\( S \)
Our Proposal: MaxUtility
MaxUtility_Diversify(k)

MaxUtility_Diversify(k): Given a query q, the set $R_q$ of results for q, two probability distributions $P(d|q)$ and $P(q'|q) \forall q' \in S_q$ measuring, respectively, the likelihood of document d being observed given q, and the likelihood of having q' as a specialization of q, the utilities $\tilde{U}(d|R_{q'})$ of documents, a mixing parameter $\lambda \in [0, 1]$, and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$\tilde{U}(S|q) = \sum_{d \in S} \sum_{q' \in S_q} (1 - \lambda)P(d|q) + \lambda P(q'|q) \tilde{U}(d|R_{q'})$$

with the constraints that every specialization is covered proportionally to its probability. Formally, let $R_q \bowtie q' = \{d \in R_q | U(d|R_{q'}) > 0\}$. We require that for each $q' \in S_q$, $|R_q \bowtie q'| \geq [k \cdot P(q'|q)]$. 
Why it is Efficient?

• By using a simple arithmetic argument we can show that:

\[
\tilde{U}(S|q) = (1 - \lambda)|S_q| \sum_{d \in S} P(d|q) + \\
+ \lambda \sum_{q' \in S_q} P(q'|q) \sum_{d \in S} \tilde{U}(d|R_{q'})
\]

• Therefore we can find the optimal set \( S \) of diversified documents by using a sort-based approach.
Algorithm OptSelect \((q, S_q, R_q, k)\)

01. \(S \leftarrow \emptyset;\)  
/* Heap(n) instantiates a new \(n\)-size heap */  
02. \(M \leftarrow \text{new Heap}(k);\)  
03. For Each \(q' \in S_q\) Do  
04. \(M_{q'} \leftarrow \text{new Heap}([k \cdot P(q'|q)] + 1);\)  
05. For Each \(d \in R_q\) Do  
06. If \(\tilde{U}(d|R_{q'}) > 0\) Then \(M_{q'}\).push(d); Else \(M\).push(d);  
07. For Each \(q' \in S_q\) s.t. \(M_{q'} \neq \emptyset\) Do  
08. \(x \leftarrow \text{pop d with the max } \tilde{U}(d|q)\) from \(M_{q'}\);  
09. \(S \leftarrow S \cup \{x\};\)  
10. While \(|S| < k\) Do  
11. \(x \leftarrow \text{pop d with the max } \tilde{U}(d|q)\) from \(M\);  
12. \(S \leftarrow S \cup \{x\};\)  
13. Return \((S);\)
**Algorithm OptSelect** $(q, S_q, R_q, k)$

01. $S \leftarrow \emptyset$;  
02. $M \leftarrow \text{new Heap}(k)$;  
03. **For Each** $q' \in S_q$ **Do**  
04. \hspace{1em} $M_{q'} \leftarrow \text{new Heap}([k \cdot P(q'|q)])$;  
05. **For Each** $d \in R_{q'}$ **Do**  
06. \hspace{1em} If $\tilde{U}(d|R_{q'}) > 0$ Then $M_{q'}.push(d)$  
07. **For Each** $q' \in S_q$ s.t. $M_{q'} \neq \emptyset$ **Do**  
08. \hspace{1em} $x \leftarrow \text{pop} d$ with the max $\tilde{U}(d|q)$ from $M_{q'}$;  
09. \hspace{1em} $S \leftarrow S \cup \{x\}$;  
10. **While** $|S| < k$ **Do**  
11. \hspace{1em} $x \leftarrow \text{pop} d$ with the max $\tilde{U}(d|q)$ from $M$;  
12. \hspace{1em} $S \leftarrow S \cup \{x\}$;  
13. **Return** $(S)$;

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>IASelect</td>
<td>$O(nk)$</td>
</tr>
<tr>
<td>xQuAD</td>
<td>$O(nk)$</td>
</tr>
<tr>
<td>OptSelect</td>
<td>$O(n \log_2 k)$</td>
</tr>
</tbody>
</table>
The Specialization Set $S_q$

- It is crucial for OptSelect to have the set of specialization available for each query.

- Our method is, thus, query log-based.

- We use a query recommender system to obtain a set of queries from which $S_q$ is built by including the most popular (i.e., freq. in query log > $f(q) / s$) recommendations:

```
Algorithm AmbiguousQueryDetect(q, A, f(), s)

/* given the submitted query q, a query recommendation algorithm A, and an integer s compute the set $\hat{S}_q \subseteq Q$ of possible specializations of q */
1. $\hat{S}_q \leftarrow A(q)$;
/* select from $\hat{S}_q$ the most popular specializations */
2. $S_q \leftarrow \{ q' \in \hat{S}_q | f(q') \geq \frac{f(q)}{s} \}$;
3. If $|S_q| \geq 2$ Then Return $(S_q)$; Else Return $(\emptyset)$;
```

D. Broccolo, L. Marcon, F.M. Nardini, R. Perego, F. Silvestri
Generating Suggestions for Queries in the Long Tail with an Inverted Index
Information Processing & Management, August 2011
Probability Estimation

\[ P(q' \mid q) = \frac{f(q')}{\sum_{q' \in S_q} f(q')} \]
Usefulness of a Result

**DEFINITION (RESULTS’ UTILITY).** The utility of a result \( d \in R_q \) for a specialization \( q' \) is defined as:

\[
U(d|R_q') = \sum_{d' \in R_q'} \frac{1 - \delta(d, d')}{{\text{rank}}(d', R_q')}.
\]

where \( R_q' \) is the list of results that the search engine returned for specialized query \( q' \).
**Usefulness of a Result**

**Definition**  (Results’ Utility). The utility of a result $d \in R_q$ for a specialization $q'$ is defined as:

$$U(d|R_q') = \sum_{d' \in R_q'} \frac{1 - \delta(d, d')}{\text{rank}(d', R_q')}.$$

where $R_q'$ is the list of results that the search engine returned for specialized query $q'$. 

$$\delta(d_1, d_2) = 1 - \cosine(d_1, d_2)$$
Experiments: Settings

- TREC 2009 Web track's Diversity Task framework:
  - ClueWeb-B, the subset of the TREC ClueWeb09 dataset
  - The 50 topics (i.e., queries) provided by TREC
  - We evaluate $\alpha$-NDCG and IA-P
- All the tests were conducted on a Intel Core 2 Quad PC with 8Gb of RAM and Ubuntu Linux 9.10 (kernel 2.6.31-22).
## Experiments: Quality

<table>
<thead>
<tr>
<th></th>
<th>( c )</th>
<th>( \alpha \text{-NDCG} )</th>
<th>IA-P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( @5 )</td>
<td>( @10 )</td>
<td>( @20 )</td>
</tr>
<tr>
<td><strong>DPH Baseline</strong></td>
<td>-</td>
<td>0.213</td>
<td>0.212</td>
</tr>
<tr>
<td><strong>OptSelect</strong></td>
<td>0</td>
<td>0.213</td>
<td>0.227</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.213</td>
<td>0.228</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.195</td>
<td>0.220</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.190</td>
<td>0.216</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td><strong>0.214</strong></td>
<td><strong>0.214</strong></td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.190</td>
<td>0.213</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.186</td>
<td>0.206</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.186</td>
<td>0.208</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.190</td>
<td>0.212</td>
</tr>
<tr>
<td><strong>xQuAD</strong></td>
<td>0</td>
<td>0.211</td>
<td>0.241</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td><strong>0.214</strong></td>
<td><strong>0.242</strong></td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.193</td>
<td>0.226</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.200</td>
<td>0.227</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.204</td>
<td>0.234</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.181</td>
<td>0.211</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.185</td>
<td>0.209</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.190</td>
<td>0.212</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.190</td>
<td>0.212</td>
</tr>
<tr>
<td><strong>IASelect</strong></td>
<td>0</td>
<td>0.206</td>
<td>0.215</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.205</td>
<td>0.214</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.193</td>
<td>0.200</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.169</td>
<td>0.185</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.182</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.198</td>
<td>0.214</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.192</td>
<td>0.208</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.192</td>
<td>0.214</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.190</td>
<td>0.212</td>
</tr>
</tbody>
</table>
## Experiments: Quality

<table>
<thead>
<tr>
<th></th>
<th>$\alpha$-NDCG</th>
<th>IA-P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@5</td>
<td>@10</td>
</tr>
<tr>
<td>DPH Baseline</td>
<td>-</td>
<td>0.190</td>
</tr>
<tr>
<td>OptSelect</td>
<td>0.213</td>
<td>0.228</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.190</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.186</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.186</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.214</td>
</tr>
<tr>
<td>xQuAD</td>
<td>0.214</td>
<td>0.242</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.214</td>
</tr>
<tr>
<td>IASelect</td>
<td>0.206</td>
<td>0.215</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.205</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.193</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.169</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.182</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.198</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.192</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.192</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.190</td>
</tr>
</tbody>
</table>
## Experiments: Quality

<table>
<thead>
<tr>
<th></th>
<th>$c$</th>
<th>$\alpha$-NDCG @5</th>
<th>$\alpha$-NDCG @10</th>
<th>$\alpha$-NDCG @20</th>
<th>$\alpha$-NDCG @100</th>
<th>$\alpha$-NDCG @1000</th>
<th>IA-P @5</th>
<th>IA-P @10</th>
<th>IA-P @20</th>
<th>IA-P @100</th>
<th>IA-P @1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPH Baseline</td>
<td>-</td>
<td>0.213</td>
<td>0.227</td>
<td>0.255</td>
<td>0.318</td>
<td>0.352</td>
<td>0.092</td>
<td>0.093</td>
<td>0.088</td>
<td>0.058</td>
<td>0.006</td>
</tr>
<tr>
<td>OptSelect</td>
<td>0</td>
<td>0.213</td>
<td>0.228</td>
<td>0.256</td>
<td>0.319</td>
<td>0.352</td>
<td>0.112</td>
<td>0.101</td>
<td>0.091</td>
<td>0.061</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.195</td>
<td>0.220</td>
<td>0.246</td>
<td>0.312</td>
<td>0.343</td>
<td>0.102</td>
<td>0.097</td>
<td>0.090</td>
<td>0.062</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.190</td>
<td>0.216</td>
<td>0.246</td>
<td>0.305</td>
<td>0.341</td>
<td>0.101</td>
<td>0.098</td>
<td>0.090</td>
<td>0.061</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.214</td>
<td>0.241</td>
<td>0.262</td>
<td>0.324</td>
<td>0.359</td>
<td>0.110</td>
<td>0.101</td>
<td>0.099</td>
<td>0.060</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.190</td>
<td>0.213</td>
<td>0.258</td>
<td>0.305</td>
<td>0.339</td>
<td>0.095</td>
<td>0.098</td>
<td>0.087</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.186</td>
<td>0.206</td>
<td>0.235</td>
<td>0.302</td>
<td>0.335</td>
<td>0.089</td>
<td>0.090</td>
<td>0.086</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.186</td>
<td>0.208</td>
<td>0.236</td>
<td>0.300</td>
<td>0.334</td>
<td>0.091</td>
<td>0.091</td>
<td>0.087</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.190</td>
<td>0.212</td>
<td>0.240</td>
<td>0.303</td>
<td>0.337</td>
<td>0.092</td>
<td>0.093</td>
<td>0.088</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td>xQuAD</td>
<td>0</td>
<td>0.211</td>
<td>0.241</td>
<td>0.260</td>
<td>0.320</td>
<td>0.354</td>
<td>0.103</td>
<td>0.102</td>
<td>0.090</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.214</td>
<td>0.242</td>
<td>0.260</td>
<td>0.323</td>
<td>0.355</td>
<td>0.108</td>
<td>0.103</td>
<td>0.089</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.193</td>
<td>0.226</td>
<td>0.249</td>
<td>0.308</td>
<td>0.341</td>
<td>0.101</td>
<td>0.101</td>
<td>0.090</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.200</td>
<td>0.227</td>
<td>0.253</td>
<td>0.315</td>
<td>0.348</td>
<td>0.099</td>
<td>0.095</td>
<td>0.087</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.204</td>
<td>0.234</td>
<td>0.262</td>
<td>0.321</td>
<td>0.354</td>
<td>0.096</td>
<td>0.099</td>
<td>0.087</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.211</td>
<td>0.236</td>
<td>0.303</td>
<td>0.336</td>
<td>0.090</td>
<td>0.095</td>
<td>0.085</td>
<td>0.058</td>
<td>0.012</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.185</td>
<td>0.239</td>
<td>0.302</td>
<td>0.335</td>
<td>0.091</td>
<td>0.092</td>
<td>0.088</td>
<td>0.058</td>
<td>0.012</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.190</td>
<td>0.240</td>
<td>0.303</td>
<td>0.336</td>
<td>0.092</td>
<td>0.093</td>
<td>0.087</td>
<td>0.058</td>
<td>0.012</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.190</td>
<td>0.240</td>
<td>0.303</td>
<td>0.337</td>
<td>0.092</td>
<td>0.093</td>
<td>0.088</td>
<td>0.058</td>
<td>0.012</td>
<td></td>
</tr>
<tr>
<td>IASelect</td>
<td>0</td>
<td>0.206</td>
<td>0.215</td>
<td>0.245</td>
<td>0.302</td>
<td>0.334</td>
<td>0.097</td>
<td>0.089</td>
<td>0.079</td>
<td>0.044</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.205</td>
<td>0.214</td>
<td>0.243</td>
<td>0.299</td>
<td>0.330</td>
<td>0.098</td>
<td>0.090</td>
<td>0.078</td>
<td>0.044</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.193</td>
<td>0.200</td>
<td>0.227</td>
<td>0.279</td>
<td>0.309</td>
<td>0.098</td>
<td>0.088</td>
<td>0.075</td>
<td>0.039</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.169</td>
<td>0.185</td>
<td>0.207</td>
<td>0.259</td>
<td>0.288</td>
<td>0.089</td>
<td>0.078</td>
<td>0.064</td>
<td>0.039</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.182</td>
<td>0.197</td>
<td>0.229</td>
<td>0.284</td>
<td>0.314</td>
<td>0.085</td>
<td>0.074</td>
<td>0.067</td>
<td>0.046</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.198</td>
<td>0.214</td>
<td>0.243</td>
<td>0.301</td>
<td>0.332</td>
<td>0.092</td>
<td>0.083</td>
<td>0.076</td>
<td>0.052</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.192</td>
<td>0.208</td>
<td>0.241</td>
<td>0.299</td>
<td>0.332</td>
<td>0.095</td>
<td>0.093</td>
<td>0.087</td>
<td>0.057</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.192</td>
<td>0.214</td>
<td>0.243</td>
<td>0.306</td>
<td>0.338</td>
<td>0.093</td>
<td>0.091</td>
<td>0.087</td>
<td>0.058</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.190</td>
<td>0.212</td>
<td>0.240</td>
<td>0.303</td>
<td>0.337</td>
<td>0.092</td>
<td>0.093</td>
<td>0.088</td>
<td>0.058</td>
<td>0.012</td>
</tr>
</tbody>
</table>
## Experiments: Efficiency

| $|R_q|$ | $k$ | 10 | 50 | 100 | 500 | 1000 |
|---|---|---|---|---|---|---|
| **OptSelect** | | | | | | |
| 1,000 | 0.34 | 0.58 | 0.66 | 0.89 | 0.98 |
| 10,000 | 1.36 | 2.13 | 2.46 | 3.32 | 3.57 |
| 100,000 | 4.81 | 8.32 | 9.57 | 12.94 | 13.92 |
| **xQuAD** | | | | | | |
| 1,000 | 0.43 | 1.64 | 3.31 | 14.82 | 30.18 |
| 10,000 | 3.27 | 16.69 | 32.22 | 148.41 | 298.63 |
| 100,000 | 36.27 | 143.67 | 285.69 | 1,425.82 | 2,849.83 |
| **IASSelect** | | | | | | |
| 1,000 | 0.57 | 1.68 | 3.92 | 20.81 | 39.82 |
| 10,000 | 4.23 | 23.03 | 40.82 | 203.11 | 409.43 |
| 100,000 | 48.04 | 205.46 | 408.61 | 2,039.22 | 4,071.81 |
Conclusions and Future Work

• We studied the problem of search results diversification from an efficiency point of view

• We derived a diversification method (OptSelect):
  • same (or better) quality of the state of the art
  • up to 100 times faster

• Future work:
  • the exploitation of users' search history for personalizing result diversification
  • the use of click-through data to improve our effectiveness results, and
  • the study of a search architecture performing the diversification task in parallel with the document scoring phase (See DDR2011 paper)
Question Time

Franco Maria Nardini
ISTI-CNR, Pisa Italy
http://hpc.isti.cnr.it/~nardini
f.nardini@isti.cnr.it
Backup Slides
The $\alpha$-normalized discounted cumulative gain ($\alpha$-NDCG) metric balances relevance and diversity through the tuning parameter $\alpha$. The larger the value of $\alpha$, the more diversity is rewarded. In contrast, when $\alpha = 0$, only relevance is rewarded, and this metric is equivalent to the traditional NDCG.

DCG measures the usefulness, or gain, of a document based on its position in the result list.

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(1 + i)}$$

Relevance scores might not be binary (i.e., relevant, not relevant) but also indicating how relevant a result is.

NDCG is the normalized version of DCG.

More info at:

\(\alpha\text{-NDCG}\)

- The \(\alpha\)-normalized discounted cumulative gain (\(\alpha\text{-NDCG}\)) metric balances relevance and diversity through the tuning parameter \(\alpha\).
- The larger the value of \(\alpha\), the more diversity is rewarded. In contrast, when \(\alpha = 0\), only relevance is rewarded, and this metric is equivalent to the traditional NDCG.
- DCG measures the usefulness of a document based on its position in the result list.
- \[\text{DCG}_p = \sum_{i=1}^{p} \frac{2^{r_{i}} - 1}{\log_2(1 + i)}\]
- Relevance scores might not be binary (i.e., relevant, not relevant) but also indicating how relevant a result is.
- NDCG is the normalized version of DCG.
- More info at:
• Intent Aware - Precision
• As “traditional” precision measured at a certain cutoff
• Basically, precision is weighted on the probability of each intent.

More info at:

• Intent Aware - Precision

• As “traditional” precision measured at a certain cutoff

• Basically, precision is weighted on the probability of each intent.

• More info at: